GROMACS - Feature #2816

GPU offload / optimization for update&constraints, buffer ops and multi-gpu communication

12/21/2018 11:16 AM - Alan Gray

<table>
<thead>
<tr>
<th>Status</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee</td>
<td>Alan Gray</td>
</tr>
<tr>
<td>Category</td>
<td>mdrun</td>
</tr>
<tr>
<td>Target version</td>
<td>2021-infrastructure-stable</td>
</tr>
<tr>
<td>Difficulty</td>
<td>hard</td>
</tr>
</tbody>
</table>

Description
Gromacs performance can be sub-optimal on modern GPU Servers.

When running on a single GPU, most force calculations can be computed done on the device, but the buffer operations plus update & constraints are done on the host, and repeated PCI-e transfers are required. Such CPU computation and PCI-e communication comprise an increasingly significant overhead as the performance of the GPU continues to increase with each subsequent generation.

On multi-GPU the situation is ever worse because the required multi-GPU communications are routed through the CPU.

NVIDIA have developed prototype code with most common compute and communication parts now device-side, with coordinate and force PCIe transfers removed for regular timesteps. Gerrit patch 8506 introduces device-side buffer ops, and patch 8859 (based on the buffer ops patch) demonstrates the remainder of the new developments:

- **GPU Update and Constraints**
- **Device MPI: PME/PP Gather and Scatter**
  - Relatively straightforward solution using CUDA-Aware MPI
- **Device MPI: PP local/nonlocal exchanges**
  - New functionality to pack device-buffers and exchange using CUDA-aware MPI
  - Similar D2D exchanges also for Constraints Lincs part

See the attached slides for more info.

These developments show major performance improvements, but are still in prototype form, and the purpose of this issue is to track the work required to integrate properly into the master branch.

TODOs common across multiple tasks:
- Add assertions when converting between rvec types on CPU to float3 types on GPU

Subtasks:
- Feature # 2817: GPU X/F buffer ops
- Feature # 2934: GPU X Buffer ops
- Task # 3237: data types mixed up and unsafe casting
- Task # 3026: add flags for GPU force buffer op / reduction activation
- Feature # 3029: GPU force buffer ops + reduction
- Feature # 3052: GPU virial reduction/calculation
- Task # 3128: do not fall back to CPU path on energy-only steps
- Feature # 3142: centralize and clarify GPU force buffer clearing
- Task # 3170: investigate GPU f buffer ops use cases
- Task # 3037: add missing cycle counters related to buffer ops/reduction launches
- Feature # 2885: CUDA version of LINCS
- Feature # 2886: CUDA version of SETTLE
- Feature # 2887: CUDA version of Leap Frog algorithm
- Feature # 2888: CUDA Update and Constraints module
- Bug # 3163: gpuupdate / task assignment stabilization
- Task # 3167: GPU update path user documentation
- Feature # 3168: GPU update release notes
## Associated revisions

### Revision bec0fa7b - 02/21/2019 09:42 AM - Artem Zhmurov

Test for LINCS and SHAKE constraints.

This version updates the tests making the selection of the constraining algorithm more abstract. Makes it possible to use the same test routines for new implementations (e.g. CPU- or GPU-based) or (and) algorithms (e.g. LINCS or SHAKE).

Partly this is preparation for the GPU-based version of the constraints (Refs #2816).

Change-Id: lce7dfdcc6d86c04656b0a1dd4e328c5a4db8a263

### Revision 0a1aae78 - 04/28/2019 05:29 PM - Artem Zhmurov

CUDA version of LINCS constraints.

Implementation of the LINCS constraints for NVIDIA GPUs. Currently works isolated from the other parts of the code: coordinates and velocities are copied to and from GPU on every integration timestep. Part of the GPU-only loop. Loosely based on change 9162 by Alan Gray. To enable, set the environmental variable GMX_LINCS_GPU.

Limitations:
1. Works only if the constraints can be split in short uncoupled groups (currently < 256, designed for H-bonds constraints).
2. Does not change the matrix inversion order for constraints triangles.
3. Does not support free energy computations.
4. Assumes no communications between domains (i.e. assumes that there is no constraints connecting atoms from two different domains).
5. Number of thread per blocks should be a power of 2 for reduction of virial to work.

TODOs:
1. Move more data from the global memory to local.
2. Change .at() to []
3. Add sorting by the number of coupled constraints to decrease warp divergencies.
4. numAtoms should be changeable (for multi-GPU case).

Refs #2816, #2885

Change-Id: l3c975cf898053b7467bcd30459e60ce2e8852be6

### Revision 02a92f23 - 04/30/2019 05:19 PM - Artem Zhmurov

CUDA version of SETTLE algorithm with basic tests

CUDA-based GPU implementation of SETTLE. This is a part of all-GPU loop. Can work isolated from other parts of the code since coordinates are copied to (from) device before (after) SETTLE kernel call. The velocity update as well as virial evaluations can be enabled.

To enable, set GMX_SETTLE_GPU environment variable.

Limitations:
1. Does not work when domain decomposition is enabled.
2. Projection of the derivative is not implemented.
3. Not fully integrated/unified with the CPU version.

TODOs:
1. Multi-GPU case.
2. Better virial reduction. This is a more general feature, not only related to constraints.
5. More cleanup in constr.cpp needed.
6. Better unit tests.

Refs #2816, #2886

Change-Id: i218e1bf1f86a2351e189a3c27f95045c06135a4

Revision d061dec5 - 05/12/2019 01:44 PM - Artem Zhmurov
CUDA version of Leap-Frog integrator with basic tests
Part of the GPU-only loop. Current version is as a stand-alone module, with its own coordinate, velocities and forces data management. To activate, set environment variable GMX_INTEGRATE_GPU.

Limitations:
-- Only basic Leap-Frog is implemented.
-- No temperature control.
-- No pressure control.

Refs #2816, #2887
Change-Id: l439d7f5fd4f69a17ca7aaa412e242ce5e3aa5d5bd

Revision 1c8eb7c5 - 06/18/2019 11:24 AM - Artem Zhmurov
Combine CUDA Leap-Frog, LINCS and SETTLE. I.
This is the first step in combining constraints and integrator into "UpdateAndConstraints" module. The initial merge does not imply any performance optimisation or code clean-up. Hence, this patch keeps all the temporary infrastructure that was built around SETTLE, LINCS and Leap-Frog to allow them to function as separate units. In the following commits, this infrastructure will be removed and these three implementations will be more closely integrated. To enable, set GMX_UPDATE_CONSTRAIN_GPU environment variable. Note, that environment variables GMX_LINCS_GPU, GMX_SETTLE_GPU and GMX_INTEGRATE_GPU will no longer work.

Refs #2816, #2888
Change-Id: l8730aad0e6a230686fe89d1157b0da2f017bc

Revision fb7a59cd - 07/03/2019 09:42 AM - Artem Zhmurov
Combine CUDA Leap-Frog, LINCS and SETTLE. II.
Stand-alone CUDA implementations of Leap-Frog, LINCS and SETTLE required additional scaffolding for integration and testing. The most prominent part of this is the management of coordinates, velocities and forces, which is removed in this commit. Management of periodic boundary conditions and virial reduction will be removed in following commits.

Refs #2816, #2888
Change-Id: l4c565a6c7088fd8059f4e7a3cb4637cb2af79ebc

Revision 6385f296 - 07/05/2019 11:05 AM - Artem Zhmurov
Remove PImpl scaffolding from CUDA version of LINCS
The CUDA implementation of LINCS was initially introduced as a stand-alone feature. This required hiding CUDA-specific variables and subroutines into the private implementation subclass. Since the LINCS is not a part of Update and Constraints module, this is no longer required and can be removed.

Refs #2816, #2888
Change-Id: i9698224d4702dfb8d99106999335c62e83a511df
Remove PImpl scaffolding from CUDA version of SETTLE

GPU version of SETTLE was implemented as a class with private implementation so it will be possible to initialize on non-CUDA hosts. Now, the implementation can be hidden inside the Update and Constraints PImpl so that the CUDA specific types and calls can be exposed in SETTLE and private implementation is no longer needed there.

Refs #2816, #2888

Change-Id: l4c78f2629be34b42bb5f4f7d34970c3e41515691

Remove PImpl scaffolding from CUDA version of Leap-Frog

Private implementation in CUDA version of Leap-Frog was used to introduce this integrator as a stand-alone unit. Now it is merged with constraints, PImpl is no longer needed.

Refs #2816, #2888

Change-Id: lea82abe016b7e15b9be44a0e1b446e12e582d3c

Refactor Leap-Frog tests and connect them to CPU version

This introduces test data object and runners to the Leap-Frog tests, which are now connected to the CPU version of Leap-Frog. This also makes possible to include tests based on the reference values, which are needed to make sure that the temperature and(or) pressure control works fine in new implementations.

Refs. #2816, #2888.

Change-Id: ld2d934c43138889ad178a94126cab4da2895bb5a

Prepare Update and Constraints for Domain Decomposition

Initial GPU-based version of the update and constraints was not designed to run with the Domain decomposition. This introduces a couple of fixes to the memory management that should allow the module to work with the DD enabled. The memory buffers are now re-allocated at the set(...) stage, if so needed.

Refs. #2816, #2888.

Change-Id: l155884f5797252cf048a6400a2dd7b042d355b7e

Make use of reference data in integrator tests

Current version of tests is based on exactly solvable model, which does not allow for testing more sophisticated cases, including when temperature or pressure control is enabled. This commit adds the tests that are based on the reference data, which can be generated for any existing use-case.

Refs. #2816, #2887.

Change-Id: l64bb2326b0adff44be8b48449ef09cd26939ea467

Remove PImpl scaffolding from CUDA version of SETTLE

GPU version of SETTLE was implemented as a class with private implementation so it will be possible to initialize on
non-CUDA hosts. Now, the implementation can be hidden inside the Update and Constraints PImpl so that the CUDA specific types and calls can be exposed in SETTLE and private implementation is no longer needed there.

Refs #2816, #2888

Change-Id: I4c78f2629be34b42bb5f4f7d34970c3e41515691

Revision 3d35e919 - 08/16/2019 11:29 AM - Artem Zhmurov
Remove PImpl scaffolding from CUDA version of Leap-Frog

Private implementation in CUDA version of Leap-Frog was used to introduce this integrator as a stand-alone unit. Now it is merged with constraints, PImpl is no longer needed.

Refs #2816, #2888

Change-Id: lea82abe016b7e15b9be44a0e1b446e12e582d3c

Revision 039709b7 - 08/16/2019 11:29 AM - Artem Zhmurov
Prepare Update and Constraints for Domain Decomposition

Initial GPU-based version of the update and constraints was not designed to run with the Domain decomposition. This introduces a couple of fixes to the memory management that should allow the module to work with the DD enabled. The memory buffers are now re-allocated at the set(...) stage, if so needed.

Refs. #2816, #2888.

Change-Id: I155884f5797252cf048a6400a2dd7b042d355b7e

Revision 22167aee - 09/10/2019 11:01 PM - Artem Zhmurov
Making DeviceBuffer available in non-GPU builds

Having DeviceBuffer available in host-side code in all builds allows to avoid passing the void-pointers for the device-side buffers.

This is a part of preparation for the GPU version of the StatePropagatorData, needed to connect all GPU routines.

Refs. #2816.

Change-Id: I174754de72999ff5299b3dd8c8a0d05494f7f4c

Revision 21abdb3c - 09/11/2019 06:35 PM - Artem Zhmurov
Reorganize PME code:

1. Split H2D copy and spread launch
2. Add getter for the padding, required in coordinates buffer
3. Add the getter for the GPU stream

TODO: Make use of DeviceBuffer

This is a part of preparation for the GPU version of the StatePropagatorData, needed to connect all GPU routines.

Refs. #2816.

Change-Id: lc0dd621ce931f8fa66e948b5240afbddef7bfb0d

Revision 873bf080 - 09/11/2019 10:53 PM - Artem Zhmurov
Decouple GPU force buffer management from buffer ops in NBNXMM

When GPU-side buffer operations are used, the total forces on the device are accumulated in NBNXMM module in the local GPU buffer. By decoupling the CPU and GPU buffer operations and making the
force buffer into an argument for the reduction function, this commit allows to take the responsibility of the GPU forces management from the NBNXM module to the third-party instance.

This commit is refactoring of the code in preparation for the introduction of the GPU-side PropagatorStateData object.

TODO: Use DeviceBuffer when passing the PME GPU forces buffer.

Refs. #2816

Change-Id: I2a1f9d12fad3f3fb5b2ce37ca3ed3d0cb91777c468

Revision 9b682479 - 09/22/2019 11:15 AM - Artem Zhmurov
Disable GPU update/constraints when neither PME nor buffer ops are offloaded.

Using the GPU-version of update makes sense if forces are already on the GPU, i.e. if at least:
1. PME is on the GPU (there should be a copy of coordinates on a GPU in rvec format for PME spread).
2. Non-bonded interactions and buffer ops are on the GPU.

This is temporary solution, needed because the buffer ops offload switch is operated by the environment variable. More favorable behavior would be to switch on the GPU buffer ops in the second case rather then disabling the GPU update.

Refs. #2816.

Change-Id: I37a9969dd6c74dcfa41a95da13ae54d014c9ea60

Revision 092a8f68 - 10/01/2019 06:22 PM - Artem Zhmurov
StatePropagatorDataGpu object to manage GPU forces, positions and velocities buffers

In current version the positions and forces on the GPU are managed by different modules, depending of the offload scenario for a particular run. This makes management of the buffers complicated and fragile. This commit adds the object responsible for management of the GPU buffers of coordinates, forces and velocities. The object is connected to all clients that use coordinates, forces and velocities buffers, while keeping the existing logic intact where its possible.

Since the H2D and D2H copies are now done in nullptr stream, some of implicit synchronization is lost. Consequently this commit does not always work properly with newly introduced buffer ops / halo exchange features. To avoid the confusion, GPU buffer ops are disabled by the assertion. There will be a separate commit with all copies done synchronously, which will work with the buffer ops. The stream- and event-based synchronization will be introduced in the follow-up commits.

Refs. #2816.

Change-Id: I2e2ba1b6436f087d1f2ef4ff876445814a724e7

Revision 77857c59 - 10/07/2019 06:51 PM - Artem Zhmurov
Pass the GPU streams to StatePropagatorDataGpu constructor

Now the StatePropagatorDataGpu has a local copy of all GPU streams and manages the update stream. This will allow to select the specific stream for a specific copy event in the follow-ups. The update stream is now created in the constructor of the StatePropagatorDataGPU object, which is a temporary solution until there is a separate device stream manager (#3115).

Notes:
- The current implementation where StatePropagatorDataGpu is also used on PME-only ranks, where many of the streams do not exist, without any restriction on the methods which would require these streams is a weakness of the design that will be dealt with in follow-up.
- The OpenCL builds unconditionally use PME stream/context, since for these this object is only used when the initial coordinates are copied.
The update stream is created in the constructor, whereas the rest of the streams is passed as arguments. This asymmetry will be removed with introduction of the centralized management of context/streams.

Refs. #2816.

Change-Id: la9b1cabd1d3d4942dba8465c716bf644037581e7

Revision 13f5fac2 - 10/15/2019 03:08 PM - Szilárd Páll

Link GPU coordinate producer and consumer tasks

The event synchronizer indicating that coordinates are ready in the GPU is now passed to the two tasks that depend on this input: PME and X buffer ops. Both enqueue a wait on the passed event prior to kernel launch to ensure that the coordinates are ready before the kernels start executing.

On the separate PME ranks and in tests, as we use a single stream, no synchronization is necessary.

With the on-device sync in place, this change also removes the streamSynchronize call from copyCoordinatesToGpu.

Refs. #2816, #3126.

Change-Id: I3457f01f44ca6d6ad08e0118d68b1def2ab0b381b

Revision 7bbfb57c - 10/16/2019 10:21 AM - Artem Zhmurov

Link GPU force producer and consumer tasks

The GPU event synchronizer that indicates that forces are ready for a consumption is now passed to the GPU update-constraints. The update-constraints enqueue a wait on the event in the update stream before performing numerical integration and constraining. Note that the event is conditionally returned by the StatePropagatorDataGpu and indicates that either the reduction of forces on the GPU or the H2D copy is done, depending on offload scenario on a current timestep.

Refs. #2816, #2888, #3126.

Change-Id: lc12b0c55b75ec5f0c31ce500a2760fb4d5c3b91

History

#1 - 02/12/2019 11:39 AM - Gerrit Code Review Bot
Gerrit received a related patchset '26' for Issue #2816.
Uploader: Artem Zhmurov (zhmurov@gmail.com)
Change-Id: gromacs~master~Ic77f07773d440c0b07c6d4d881f3d9c8b8b520c0
Gerrit URL: https://gerrit.gromacs.org/8982

#2 - 02/20/2019 03:45 PM - Gerrit Code Review Bot
Gerrit received a related DRAFT patchset '1' for Issue #2816.
Uploader: Artem Zhmurov (zhmurov@gmail.com)
Change-Id: gromacs~master~I3c975cf898053b7467b0a1dd4e328c5afdb8a263
Gerrit URL: https://gerrit.gromacs.org/9193

#3 - 02/26/2019 02:03 PM - Alan Gray
I want to add a subtask here for "GPU Halo exchange", but can't see a way to do it. Are special permissions required?

#4 - 02/26/2019 03:12 PM - Gerrit Code Review Bot
Gerrit received a related patchset '2' for Issue #2816.
Uploader: Alan Gray (alang@nvidia.com)
Change-Id: gromacs~master~I8e6473481ad4d943df78d7019681bfa821bd5798
Gerrit URL: https://gerrit.gromacs.org/9225

#5 - 02/27/2019 04:53 PM - Gerrit Code Review Bot
Gerrit received a related DRAFT patchset '1' for Issue #2816.
Uploader: Artem Zhmurov (zhmurov@gmail.com)
Gerrit received a related DRAFT patchset '4' for Issue #2816. 
Uploader: Artem Zhmurov (zhmurov@gmail.com) 
Change-Id: gromacs~master~I218e1bf1f86a2351e189e3c27950f45c06135a4 
Gerrit URL: https://gerrit.gromacs.org/9244

We need to decouple these changes; there are several distinct features that are proposed here, so we need redmine issues for those. I would also prefer to organize trees of issues around a certain target feature-set, e.g. single-GPU no-DD all offloaded, or multi-GPU with-DD, most offloaded, etc. While feature sets may overlap, the higher-level features are these parallelization functionalities that will depend/be related to both common and individual tasks. Consequently, at least a separate LINCS, SETTLE, Update, halo exchange, and PP-PME comm issues would be desirable, possibly even separate ones for with/without communication (when this makes sense).

Yes. I already tried to create a sub-task here for halo exchange, but couldn't see how to do it. Could you let me know how you did it for the "GPU X/F Buffer Ops" task? It may be a permissions thing.

I've created blank features for the GPU-only loop. Will start filling them up.

Gerrit received a related DRAFT patchset '2' for Issue #2816. 
Uploader: Artem Zhmurov (zhmurov@gmail.com) 
Change-Id: gromacs~master~I4c65a6c7088fd8059f4e7fa3cb4637cb2af79e9bc 
Gerrit URL: https://gerrit.gromacs.org/9349

- Subject changed from Device-side update&constrains, buffer ops and multi-gpu comms to GPU offload / optimization for update&constrains, buffer ops and multi-gpu communication
- Description updated

Core parts are enabled in 2020, but bumping this parent task to 2021 for follow-up subtasks.

Files
NVDevUpdate21Dec18.pdf  1.21 MB  12/21/2018  Alan Gray

Files
NVDevUpdate21Dec18.pdf  1.21 MB  12/21/2018  Alan Gray